REMARKS

Claims 1-13, 16, 18-28, 31, 33, 36, 38-49, and 51-54 were pending when the Office Action was mailed. Applicants herein amend claims 1, 6, 7, 18, 33, 41, 45, 49, and 54 and cancel claim 48. Accordingly claims 1-13, 16, 18-28, 31, 33, 36, 38-47, 49, and 51-54 are currently pending.

Applicants would like to thank the Examiner for indicating that claims 41-44 and 54 would be allowable if rewritten in independent form. Applicants have rewritten claims 41 and 54 in independent form and respectfully request that the Examiner withdraw the rejections of these claims and claims 42-44, which depend from claim 41.

The Office Action rejects claims 1-13, 16, 18-28, 31, 33, 36, 38-40, 45-49, and 51-53 under 35 U.S.C. § 103(a) over the combination of U.S. Patent No. 6,341,301 to Hagan ("Hagan") and U.S. Patent No. 6,209,003 to Mattis et al. ("Mattis"). Applicants respectfully traverse these rejections. Nevertheless, applicants herein amend the claims to clarify the subject matter for which protection is sought.

Applicants' technology enables threads in a multithreaded environment to access a single collection of items in parallel. (Specification at Abstract, ¶ [0082]). The disclosed technology manages the collection using a write counter, a read counter, a lower bound, a bucket array, and a data structure (e.g., linked list) associated with each bucket. (Specification at ¶ [0082], Figure 12). The read and write counters reference the buckets to which the next read or write will occur and the lower bound indicates a lower bound on the number of items in the collection. (Specification at ¶ [0082]). Threads add and remove items from the bucket array in a circular manner. (*Id.*). That is, the j^{th} item added to the collection is added to bucket[j modulo N] and the k^{th} item read from the collection is read from bucket[k modulo N], where N is the number of buckets in the array. (*Id.*). Access to each bucket is synchronized using a synchronization mode of sync and a full/empty bit associated with each bucket. When a bucket's full/empty bit is set to full and a thread attempts to read from the bucket using a synchronization access mode of sync, the thread is permitted to access the bucket and the bit is set to empty. (Specification, ¶¶ [0084]-[0085]). However, if a thread attempts a read of a bucket whose full/empty bit is set to empty, the thread is blocked from accessing the bucket until the bit is set to full. In contrast, when a thread attempts a synchronized write of a bucket whose full/empty bit is set to full, the thread is blocked from accessing the bucket. When a thread attempts a synchronized write of a bucket whose full/empty bit is set to empty, the thread is permitted to write to the bucket and the bit is set to full. In short, the full/empty bit indicates whether a bucket is available for consumption or available to be written to. Applicants' technology enables parallel access to the collection by multiple threads by effectively locking only a portion of the collection (i.e., represented by a bucket) at a time while the remainder of the collection remains accessible by other threads. Thus, each bucket can be locked and accessed by different threads at the same time using a synchronization access mode of sync. This allows different threads to access different portions of the collection without waiting for other threads to finish their access.

Hagan is directed to techniques for handling a plurality of queues within a data processing system. (Hagan, Abstract; 2:2-8). Each queue has an associated access pointer and maintains a separate collection of tasks corresponding to a specific priority level; tasks of one queue are processed only after each task of a higher priority level queue is processed. (Hagan, 6:10-26). In the event of a context switch (i.e., a condition requiring a change in queues), the access pointer is stored and another queue's access pointer is designated as the current pointer and used to process entries in its associated queue. In this manner, Hagan processes queues exclusively to ensure "that only one queue is being processed while others remained locked." (Hagan, 8:61-64).

Mattis describes an I/O core of a traffic server for adjusting the I/O rate of the traffic server to match the transmission speed of the link between a client and the Internet. (Mattis, 7:17-40). Mattis stores connections to the traffic server in buckets and polls connections during successive examination of each bucket to determine the amount of information that has accumulated in a buffer associated with the connection and to adjust a period value for the connection based on the accumulated information. (Mattis, 7:40-47). Each connection "is then stored in a different bucket [based on] the sum of the current bucket number and the period value." (7:47-49).

Each claim now recites that different portions of a collection of data items can be accessed in parallel. For example, claim 1 now recites "each bucket in the bucket array corresponding to a portion of the collection of data items" and "wherein multiple threads can be simultaneously accessing different buckets." Claim 18 now recites "each bucket in the bucket array corresponding to a portion of the collection of data items, wherein data items associated with different buckets can be accessed simultaneously." Claim 33 now recites "each bucket in the bucket array corresponding to a portion of the collection of data items, wherein multiple readers and writers can be accessing data items of different buckets simultaneously." Claim 49 now recites "each bucket corresponding to a portion of a collection of data items, wherein multiple readers and writers can access the data items of different buckets in parallel."

Neither Hagan nor Mattis disclose these features. According to the Office Action, each of Hagan's queues corresponds to applicants' buckets. However, Hagan's queues are only accessed one at a time, not simultaneously. Hagan's common queue handling routine "manage[s] any number of queues in a mutually exclusive fashion" to ensure that only one queue is processed at a time. (Hagan, 8:48-49, 61-64). While Hagan's technique is processing one queue, entries of other queues are not processed until a context switch to another queue is performed. (Hagan, 6:10-11). In Mattis, buckets storing connections are "successively examined" to gather information about connections stored in the buckets. (Mattis, 41-45). Thus, in examining its set of buckets, Mattis's technique accesses each bucket one by one, not simultaneously. Accordingly, both Hagan and Mattis fail to teach or suggest accessing different portions of a collection of items simultaneously, as recited by claims 1, 18, 33, and 49.

Accordingly, claims 1, 18, 33, and 49 are patentable over the applied references, as are their dependent claims. Applicants respectfully request that the Examiner reconsider and withdraw the rejections of these claims.

In view of the above amendments and remarks, applicants believe the pending application is in condition for allowance and respectfully request reconsideration.

Please charge any deficiency in fees or credit any overpayment to our Deposit Account No. 50-0665, under Order No. 324758003US6 from which the undersigned is authorized to draw.

Dated: October 8, 2008

Respectfully submitted,

By Maurice J Pirio

Maurice J. Pirio
Registration No.: 33,273
PERKINS COIE LLP
P.O. Box 1247
Seattle, Washington 98111-1247
(206) 359-8548
(206) 359-9000 (Fax)
Attorney for Applicant